Power Analysis of Synchronous Stream Ciphers with Resynchronization Mechanism *

Joseph Lano**, Nele Mentens, Bart Preneel and Ingrid Verbauwhede

Katholieke Universiteit Leuven, Dept. Elect. Eng.-ESAT/SCD-COSIC,
Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium
{joseph.lano,nele.mentens,bart.preneel,ingrid.verbauwhede}@esat.kuleuven.ac.be

Abstract. In this paper we discuss power analysis of stream ciphers. In such attacks, one measures the power consumption of the algorithm and tries to extract the secret key from these measurements. Power attacks have been mounted against block ciphers and public key algorithms but not yet against stream ciphers. In this paper we give a theoretical framework that shows that power analysis of stream ciphers with resynchronization mechanism is feasible and describe possible attack methodologies against $A5/1$ and $E0$.

1 Introduction

Research towards the security of cryptographic algorithms has been essentially concerned with classical cryptanalysis. The algorithm is regarded as an abstract mathematical model on which a variety of attacks are investigated.

In reality an algorithm will always be implemented and will run on some specific hardware. Hence, an attacker can also mount an attack on the implementation of the algorithm. Even if the cipher is mathematically sound, he may in such way be able to recover the secret key. Such attacks are often called side-channel attacks. We can roughly divide these attacks into active attacks and passive attacks. In active attacks one tries to deviate from the proper functioning of the device, for instance by fault induction. Passive attacks only require the observation of the side-channels of the device, which functions normally. Passive attacks include timing analysis, power analysis, and electromagnetic analysis.

Concerning symmetric algorithms, most research has been directed towards attacks on block ciphers, namely DES and the AES finalists. Very little research has been done on the resistance of stream ciphers against side-channel attacks. An example of such research is a theoretical discussion on fault analysis, see [5].

In this paper we will develop a theoretical framework for the power analysis of synchronous stream ciphers with resynchronization mechanism. Power analysis attacks measure the power consumption of the device and try to extract the

* This work was supported by the Concerted Research Action (GOA) Mefisto-2000/04 of the Flemish Government.

** Research financed by a Ph.D. grant of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen)

secret key of the algorithm during its operation. Two types of analysis exist: the simple power analysis (SPA), and the much more powerful differential power analysis (DPA), introduced by Kocher, Jaffe and Jun in [6]. Successful DPA attacks have been implemented against block ciphers such as DES and the AES candidates and against several public-key cryptosystems.

The outline of this paper is as follows. In Sect. 2, we give a brief description of power analysis attacks. In Sect. 3, we discuss power analysis of stream ciphers and we describe two attacks: one on the irregularly clocked $A5/1$ algorithm used in GSM, and one on the regularly clocked $E0$ stream cipher from the Bluetooth standard.

2 Description of Power Analysis Attacks

2.1 Simple Power Analysis (SPA)

In simple power analysis, one measures the power consumption of the algorithm and tries to extract the secret key from this measurement. This is possible if some operations are directly dependent on the value of the key. This is for instance the case if the algorithm has some key-dependent conditional jumps, etc. Simple power analysis can normally be prevented quite easily by some implementational tricks.

2.2 Differential Power Analysis (DPA)

DPA attacks were introduced in [6]. DPA is made possible by the fact that the power consumption of an operation is influenced by the value of the bits being manipulated. This is impossible to see on a single trace (i.e., the measurement of the power consumption of the algorithm during one encryption), but by combining the knowledge obtained from many traces one is able to extract the secret key. Different consumption models have been developed that describe this effect, and although they have their limitations (see e.g. [1]), the attacks appear to work in practice.

DPA attacks require many (thousands) measurements, in which both known (usually plaintexts, ciphertexts) and unknown (usually a secret key) bits are used. The central idea of the DPA attack is to guess a small subset of key bits. The knowledge of these key bits allows us to determine the value of one bit b that appears during the computation. We then divide our measurements into two subsets depending on the value of this bit. The consumption of the system allows us, on average, to distinguish the consumption when computing with $b = 0$ from the case $b = 1$. So if we see a bias in the difference of our two sets, our guess of the key bits was correct. In case our guess of the key bits was wrong, we have divided the measurements into two random sets and do not see any bias.

3 Power Analysis of Stream Ciphers

Power analysis attacks have not been applied to synchronous stream ciphers. The problem with DPA attacks against stream ciphers is that the key stream is computed independently from the plaintext to be encrypted. So the interference between known and unknown values required for the DPA attack is not present here.

However, in many applications stream ciphers require frequent resynchronization, to prevent synchronization loss between sender and receiver. In this case, the state of the stream cipher is frequently reinitialized with the same secret key K and with a different initial value IV. This resynchronization should be highly nonlinear in order to prevent resynchronization attacks [4]. For instance the A5/1 algorithm used in GSM communications encrypts the data in packets of 224 bits, and the E0 algorithm used in Bluetooth employs packets of at most 2745 bits.

An interesting observation is that the resynchronization inserts known values into the system. This enables an attacker to do consecutive power measurements on many frames with the same secret key but with many different IVs. It seems that all conditions are present to perform a successful DPA attack on a stream cipher with reinitialization.

In this section, we will describe theoretical power analysis on two very common types of LFSR-based stream cipher. The first is the GSM algorithm $A5/1$, which is an example of an irregularly clocked stream cipher. The second is the Bluetooth algorithm $E0$, which is an example of a combiner with memory.

3.1 The $A5/1$ Stream Cipher

Description of $A5/1$. $A5/1$ consists of three LFSRs, as depicted in Fig. 1. Every 4.6 milliseconds, a frame is encrypted by A5/1 using the fixed secret key K and a frame counter F_n. The procedure for generating each frame is as follows:

1. $t = 0 \ldots 63$:The LFSRs are set to zero and then clocked 64 times. At each clock one bit of the secret key is XORed in parallel into the least significant bits of the three LFSRs.
2. $t = 64 \ldots 85$: The LFSRs are clocked 22 more times. At each clock one bit of the frame counter F_n is XORed in parallel into the least significant bits of the three LFSRs.
3. $t = 86 \ldots 413$: From now on the LFSRs are clocked irregularly as follows: at each clock a majority bit $m = majority(c_1, c_2, c_3)$ is calculated. LFSR Ri $(1 \leq i \leq 3)$ is now updated only if $m = c_i$. This implies that at each clock either 2 or 3 LFSRs are updated. In this way 328 bits of output are generated. The first 100 bits are discarded, the next 228 bits are used to encrypt the GSM conversation.

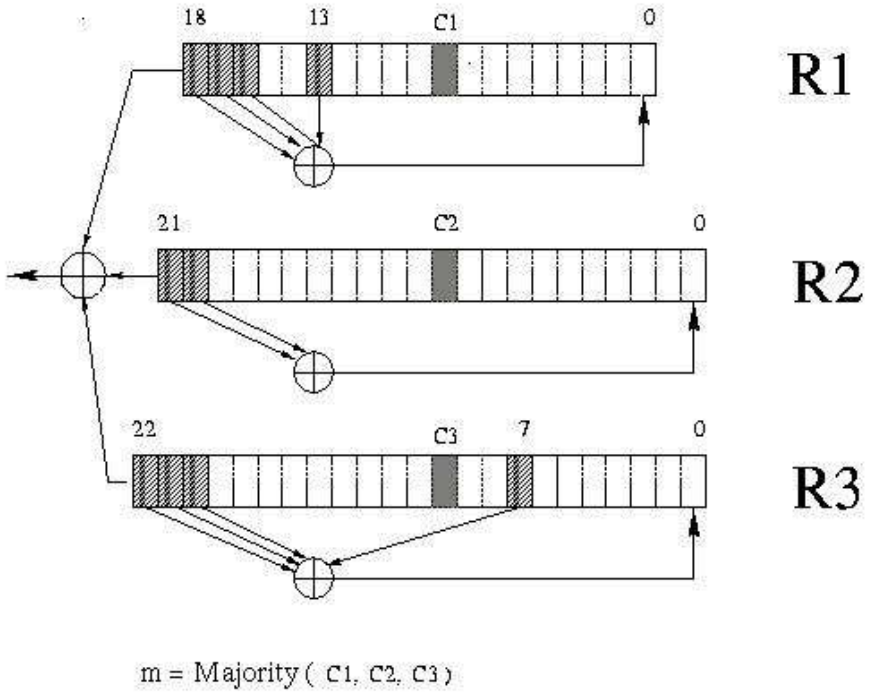

Fig. 1. A5/1

Power Analysis Attack on *A5/1* The idea for the Power Analysis is very simple: the attacker does not know whether 2 or 3 LFSRs are clocked at a time t. However, the power consumption of A5/1 while clocking 3 LFSRs in parallel can be expected to be higher than when clocking only 2 LFSRs. We will now show how to deduce the secret key from power measurements by using power analysis techniques.

Assume we have measured a large number of power traces, obtained from a single secret session key K. For example, 1000 traces would correspond to about 4.6 seconds of telephone conversation.

We now focus first on $t = 86$, the time of the first irregular clock. For each trace i, we know that the three bits used for the majority are a combination of unknown but constant key material and known frame-dependent values as follows:

$$\begin{cases} c_1^i = k_1 \oplus iv_1^i \\ c_2^i = k_2 \oplus iv_2^i \\ c_3^i = k_3 \oplus iv_3^i \end{cases} \tag{1}$$

Now we proceed as follows: we guess the triplet (k_1, k_2, k_3). Based on this guess, we divide our power traces into two sets S_2 and S_3. A trace goes into set S_3 if the

three values in (1) are equal and into S_2 otherwise. If our guess was correct, all traces in S_3 represent instances that had 3 LFSRs clocked at $t = 86$, and all traces in S_2 had only 2 LFSRs clocked at $t = 86$. By computing $mean(S_3) - mean(S_2)$, we can thus expect to see a significant peak in this differential trace. If the guess were not correct, our division into two sets will be wrong and the difference between the two means will be much lower. Note that our division into sets will also be correct for the complement of the triplet, so we will retain two possible values for the three bits. In other words, we have learnt 2 bits of information on the key.

Now that we know which LFSRs have been updated at $t = 86$, we can calculate which bits are in the majority bits positions at $t = 87$. By continuing to work iteratively we can deduce the secret key of A5/1.

3.2 The *E*0 Stream Cipher

Description of *E*0. $E0$ is the stream cipher used for encryption in the well-known Bluetooth standard [2]. For a full description of $E0$, we refer to [2], Vol. 2, pp. 763-772. A working implementation in C of $E0$ can be found at [7].

When two Bluetooth devices want to communicate, they undergo a key exchange protocol whereby they agree on a 128-bit encryption key K_C (it may be smaller than 128 bits, but we will only describe the 128-bit case here). When encrypting a packet, K_C is linearly combined with the 48-bit Bluetooth address of the master device and with the 26-bit clock to form the initial state of the LFSRs for the two-level key stream generator.

The general design of E0 is inspired by the summation combiner of Rueppel. It is depicted in Fig. 2. E0 consists of 4 LFSRs, with total length of 128 bits (25 bits for LFSR1, 31 for LFSR2, 33 for LFSR3, 39 for LFSR4). E0 also contains 4 nonlinear memory bits, the so-called blender. At each iteration the LFSRs are clocked, the new state of the blender is calculated as a nonlinear function of its current state and of the 4 output bits of the LFSRs, and one bit of key stream is calculated as the XOR of the 4 output bits of the LFSR and of the one output bit of the blender.

As said, in a first level the LFSRs are linearly loaded with the key, the address and the clock. The blender is set to zero. Then, E0 is clocked 200 times. The first 72 outputs are discarded, the next 128 outputs will be the initial state of the LFSRs in the second level. The initial state of the blender in the second level is set to equal the final state of the blender after the first level.

A DPA attack on *E*0. The case of the two-level key stream generator is interesting for power analysis: the linearly loaded first level key stream generator is shielded from an attacker who tries to do a resynchronization attack, but it remains exposed to an attacker who is doing a SCA attack.

In the DPA attack, our aim is to recover the encryption key K_C. In a first phase, power measurements have been performed for N (typically 1000) frames, each encrypted with the same K_C but with a different (known) clock counter value.

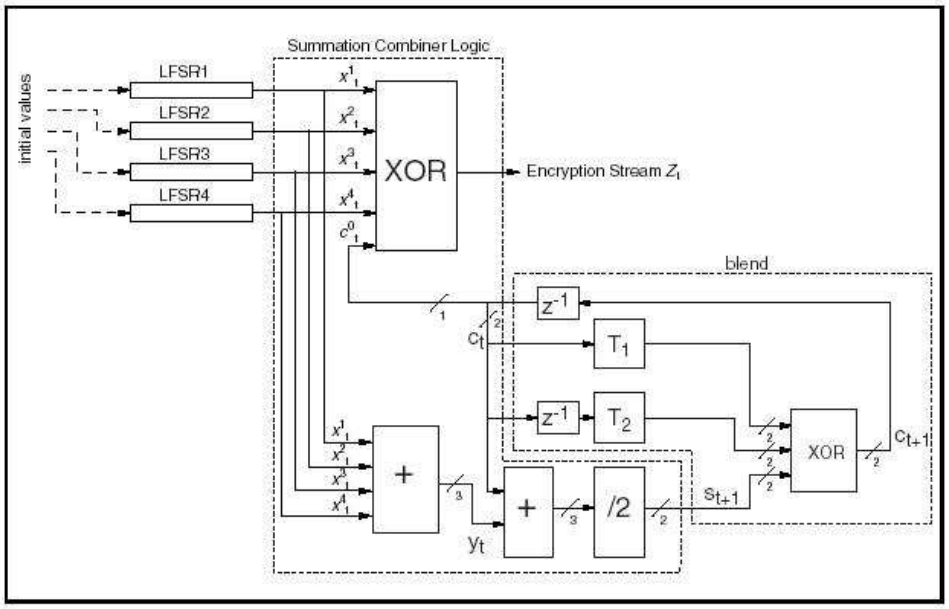

Fig. 2. E0

To simplify the description of the DPA attack, we will for now assume that the initial value of each bit of the LFSRs is influenced by the resynchronization. This is not the case in $E0$, but the attack can be adapted quite easily to circumvent this.

In this first simple attack, we assume the hamming weight model: the power consumption is correlated positively with the hamming weight of the inputs. We focus on the XOR output function, which takes as inputs the five bits x_t^1, x_t^2, x_t^3, x_t^4 and c_t^0. In the first iteration of level 1, we know that c_1^0 is equal to zero. This is due to the structure of the blender update function. For each frame i, we know the initial values involved in this first iteration. We thus get the following equations:

$$\begin{cases} x_1^1 = k^1 \oplus iv_i^1 \\ x_1^2 = k^2 \oplus iv_i^2 \\ x_1^3 = k^3 \oplus iv_i^3 \\ x_1^4 = k^4 \oplus iv_i^4 \end{cases} \tag{2}$$

We now guess the four bits k^1, k^2, k^3 and k^4. Based on this guess, we divide our power traces into five sets S_0, S_1, S_2, S_3 and S_4. For instance, set S_0 contains all measurements for which the tuple $(x^1x^2x^3x^4)$ has hamming weight 0 for the current guess of the key. We then compute the mean of these sets, and the differences of these means. It can be expected that the guess of the four key bits for which the bias detected is largest will be the correct guess of these four key bits.

By knowing the four key bits, we can now calculate the next state of the blender. We thus know c_2^0 for all our measurements, and we can again perform the DPA attack on the second iteration of the first level. We continue proceeding iteratively until we have recovered the entire key K_C.

Many other attack scenarios are imaginable. For instance, one could focus on the update functions of the individual LFSR during the first level and perform a similar attack. As the update function of each LFSR has weight 5, the complexity of this approach should not be higher. It would also not require to keep track of the blender state. A third alternative would be to attack the addition performed at the entrance of the blender. In all scenarios the attack would be similar as described above.

4 Conclusion

The field of side-channel analysis is a very interesting area of research. Unfortunately, very little attention has been given to the application of SCA on stream ciphers.

In this paper, we have given the first theoretical application of power analysis on stream ciphers. We hope to test the practicality of these attacks in the coming weeks. Please refer to [3] for the latest version of this paper. We hope that this paper can stimulate further research on side channel analysis of stream ciphers.

References

1. M. Akkar, R. Bevan, P. Dischamp, D. Moyart, *Power Analysis, what is now possible...*, Asiacrypt 2000, LNCS 1976, T. Okamoto, Ed., Springer-Verlag, pp. 489–516, 2000.
2. Bluetooth S.I.G., *Specification of the Bluetooth System, Version 1.2.*, available from www.bluetooth.org/spec, 2003.
3. COSIC publication database, https://www.cosic.esat.kuleuven.ac.be/publications/
4. J. Daemen, R. Govaerts, J. Vandewalle, *Resynchronization Weaknesses in Synchronous Stream Ciphers,* Eurocrypt 1993, LNCS 765, T. Helleseth, Ed., Springer-Verlag, pp. 159–167, 1993.
5. J. Hoch, A. Shamir, *Fault Analysis of Stream Ciphers*, CHES 2004, LNCS 3156, M. Joye, J. Quisquater, Eds., Springer-Verlag, pp. 240–253, 2004.
6. P. Kocher, J. Jaffe, B. Jun, *Differential Power Analysis*, Crypto'99, LNCS 1666, M. Wiener, Ed., Springer-Verlag, pp. 388–397, 1999.
7. M. Saarinen, *e0.c*, available from www.cc.jyu.fi/ mjos/e0.c, 2000.